ÖREBRO UNIVERSITET

INSTITUTIONEN FÖR
NATURVETENSKAP OCH TEKNIK

Inbyggda System, DT511G

Inbyggda System, teori

A001

4,5 högskolepoäng

Skriftlig tentamen

2023-08-14

Inbyggda system, DT511G (A001)

Tillåtna hjälpmedel: penna, radergummi, engelska-svenska ordbok, miniräknare

Instruktioner:

Läs igenom alla frågor noga.
Ange tentamenskoden på svarsdokumentet.
Du kan svara på *Svenska* eller *Engelska*.
***Skriv tydligt* (gäller även för en digital tentamen).**
Detta är en individuell examination - alla misstankar om otillåtet samarbete kommer att rapporteras.
Ansvarig lärare finns tillgänglig via telefon fr.o.m. andra skrivtimmen.
Skriv läsligt!
förklara och motivera era svar

Ansvarig lärare: Pascal Rebreyend, tel: 0702001422

För betyg G krävs 50% av total poäng (20 på 40)
(26 poång gav betyg 4, och 32 poäng gav betyg 5)

Lycka till!

Question 1 (3 points) (interrupt freertos non isp)

You want to recruit a new employee to develop embedded softwares on arduino-like board. During the recruitment process you ask a candidate to perform a task with an interrupt and the candidate produce the following part. How you will analyze the code? Do you plan to recruit this candidate and why? *(... in the code means lines useless for the purpose of this question)*

```
#include "FreeRTOS_ARM.h"
...

void setup( void )
{
...
    attachInterrupt(inputPin, MyInterRupt, RISING);
  ...
}

void  MyInterRupt( void)
{
  xSemaphoreTake( BinarySemaphore1, 500);
  glob2=glob1+25;
  glob1=glob1+1;
  digitalWrite(13, glob2);
  xSemaphoreGive( BinarySemaphore1);
}
...
```

Question 2 (5 points)

To avoid deadlocks and long delays when interrupts or high priority tasks are activated , an embedded system may alter the priority of some tasks. What are the different options which can be used?

Question 3 (7 points) (timers, programming)

In many sports or games players have a limited time to play or to do an action. Therefore, your company want to develop a clock to clearly show the remaining time before the play or action has to been done. The clock is composed of two rings of 60 leds each. The device has also 2 switches to control it. The clock is able to display remaining time up to one hour.

The user manual of the device contains the following:

- To start to display the countdown, press the button A. If the remaining time is more than one minute, the outer ring of leds will represent the number of minutes remaining and the inner ring the number of seconds remaining. (note: the number of leds which are on equals the number to be represented, and leds are lighted anti-clockwise from the top). If the remaining time is 60 seconds or less, the leds of the outer rings will represent the remaining time. If the time is less than 10 seconds, some (you choose) leds of the inner ring will flash with a frequency which increase when the number of seconds remaining is decreasing.
- When the remaining time reached 0, all leds are off.
- By pressing the button B, you can select the duration of the countdown timer. The sequence of possible choices is 60 minutes, 45 minutes, 5 minutes, 60 seconds, 30 seconds and 15 seconds. Each choice is represented by a led pattern (the outer ring for the minutes, and the inner one for the seconds). The first press on B is showing the current choice. If B is pressed again in less than 10 seconds the next choice will be selected. If B is not pressed within 10 seconds, the last choice is memorized and the device go back to a waiting state. (all leds off, waiting for a button to be pressed)

You can use a specific library to turn on or off the different leds by using the function *Turn_Le*d(ID,b) where ID is a number indicating which led should be updated (from 0 included to 119) and b is a boolean (true means led is lighting, false the led is off).

You can use at the maximum 2 timers.

Describe the general layout of the code and then write the corresponding functions.

Question 4 (5 points)

You have to schedule a set of periodic tasks. The table below summarize for each task its period and its execution time. Are you able to find a feasible schedule for this set of tasks? If yes, which one? If not, can someoneelse find a feasible schedule? (justify your answer)

Task	Period	Execution time
A	17	1
B	7	2
C	28	1
D	5	2
E	14	1
F	12	1

Question 5 (3 points)

What are the differences and common things between pipes and queues in a realtime system?

Question 6 (7 points)

Messages in a queue are generally sorted in a FIFO way (First In, First Out). But in practice we may have messages with different priorities and therefore we want to sort the queue of messages in such way the priorities of the messages are kept.

In order to address this issue, write a small function which can reorder messages of a FreeRTOS queue. Messages are composed of two fields: The first is an integer indicating the priority of the message (The higher the number is, the higher the priority is) and then the payload which is another integer.

The current system works as the following:

- Some tasks are sending messages to the queue using the classical function *xQueueSendToBack()*. Every message contains one integer to indicate the priority and one integer (payload)
- Some tasks may take messages from the (front of the) queue using *xQueueReceive*.
- The length (the maximum number of messages) is the predefined value Max_Messages

With the current system, the priority information is not used in the queue and therefore a low priority message will be taken out the queue and processed even if an higher priority message exist in the queue but was entering the queue after the low priority message.

Your task is to write the code of a task doing the following in a periodic fashion:

- Reorder the messages of the queue (by taking them out and then putting them back in the queue) such as:
    - Messages will be order from the highest to the lowest priority
    - Messages of same priority will have the same order after than before the reordering

Describe how you will do it and write the corresponding code. Your new function should be safe. (no message should be lost,...)

Question 7 (7 points)

We have the following set of tasks (same arrival time) . Each task is define by its execution time, it's deadline and precedences listed in the following table. The precedences' column represents for each task which tasks should be completed before the task can start. (example: Task B depends on task A; cannot start before A is completed)

Using the EDF (Earliest Deadline First) algorithm, are you able to find a feasible schedule? (4 points)

If yes, what is this schedule? If no, do you think it's useful to try another algorithm and which one(s) you may try? (3 points)

Task	Execution time	Deadline	Precedences
A	1	3	none
B	3	4	A
C	1	5	A
D	1	6	B
E	1	7	B
F	3	9	C

(One more question on the next page)

Question 8 (3 points)

Why most of the realtime system have no virtual memory mechanism and no memory protection?